

Funzioni utili della libreria string.h

=====

Quello che segue `Ã` un elenco delle funzioni di `string.h` piÃ comunemente utilizzate. Di ciascuna `Ã` fornito il prototipo e una descrizione.

Per utilizzare le funzioni sotto elencate `Ã` necessario inserire nel programma la direttiva

```
#include <string.h>
```

Tutte le funzioni sotto descritte operano su C-string, overosia array di caratteri utilizzati per contenere stringhe terminate da `'\0'`. Per semplicitÃ , nel seguito si userÃ semplicemente il termine "stringa" per indicare una C-string.

`strlen` - fornisce la lunghezza di una stringa

~~~~~

```
size_t strlen(const char *Stringa)
```

dove `size_t Ã` un tipo di dato assimilabile ad un intero.

Restituisce la lunghezza (ovvero il numero di caratteri contenuti) della stringa puntata da `Stringa`, escluso il `'\0'`.

Ad esempio `strlen("prova")` restituisce 5.

`strcpy` - copia una stringa in un'altra

~~~~~

```
char * strcpy (char *Destinazione, const char *Origine)
```

Copia la stringa `Origine` nella stringa `Destinazione`, incluso il `'\0'` in `Origine`.

Nota: `Ã` compito del programmatore assicurarsi che `Destinazione` comprenda un numero di caratteri sufficiente ad accogliere il contenuto di `Origine`.

Restituisce il valore di `Destinazione`.

`strcmp` - compara due stringhe

~~~~~

```
int strcmp(const char *Str1, const char *Str2)
```

Compara le stringhe Str1 e Str2 carattere per carattere, iniziando dal primo carattere e fermandosi quando incontra due caratteri diversi in posizione corrispondente.

Restituisce un valore cos'è determinato:

<0 se Str1 e Str2 differiscono, e nella prima posizione in cui Str1 e Str2 contengono caratteri diversi il carattere di Str1 ha valore inferiore

=0 se Str1 e Str2 sono uguali (overosia contengono gli stessi caratteri fino al primo '\0' incluso)

>0 se Str1 e Str2 differiscono, e nella prima posizione in cui Str1 e Str2 contengono caratteri diversi il carattere di Str2 ha valore inferiore

```
strstr - cerca una stringa all'interno di un'altra stringa
```

```
char *strstr(const char *Pagliaio, const char *Ago)
```

Cerca la stringa Ago (senza considerare il '\0' di terminazione) all'interno della stringa Pagliaio. Se la trova, restituisce un puntatore all'elemento di Pagliaio che contiene il primo carattere di Ago; altrimenti restituisce NULL.

Se Ago compare piú volte in Pagliaio, viene restituito un puntatore alla prima occorrenza (quella piú vicina al primo elemento).

Di questa funzione esiste anche la versione

```
char *strcasestr(const char *Pagliaio, const char *Ago)
```

che ignora la differenza tra maiuscole e minuscole, overosia considera uguali due caratteri (uno di Ago e uno di Pagliaio) quando essi rappresentano la stessa lettera anche se uno dei due in versione maiuscola e l'altro in versione minuscola.